

November 1982

Bubble Chip Packs 4 Mbits Into 1-Mbit Space

Hudson Washburn
Design Engineer

Sam Nicolino
Design Engineer
Intel Corporation
Santa Clara, California

BehindTheCover

Right after putting their 1-Mbit bubble memory chip into production several years ago, designers at Intel decided to try various sections of what would be needed to build a 4-Mbit device. Although several were fabricated and proved functional, priorities in ironing out the production problems for the 1-Mbit chip forced them to put the 4-Mbit design on the back burner, working on it as a secondary project. Finally, though, the years of patience are paying off, and as our cover story in this issue (p. 1) highlights, the 4-Mbit magnetic bubble memory—the i7114—is functional.

Fortunately, the designers have been able to time the developments so that both the bubble chip and its associated support chips will be ready at the same time. As Mike Eisele, product manager for the Magnetic Bubble Memory Division, notes, that wasn't the case for the 1-Mbit device—it took Intel a lot longer than it expected to make the controller fully functional.

In developing the 4-Mbit memory, Hudson Washburn, design engineer, expected that the control elements on the chip—the bubble generator, transfer gates, replicator, and detector—would be the most difficult sections to get to work, whereas he thought that the propagation paths would be relatively simple to implement. But when actually trying to create the memory chip, he and the other researchers found that the control sections performed fine after only a few iterations while the propagation paths turned out to be the tricky development problem.

Additionally, mastering the technology needed to build the 4-Mbit bubble chip was a long, hard process with many half steps back, Washburn says. However, work on the 1-Mbit device also helped the bigger memory: Every time something happened that caused yield problems on the 1-Mbit chip, work was stopped on the new circuit. When the problem or problems on the 1-Mbit process were solved, the designers applied what they learned to the 4-Mbit technology.

Also, the designers decided to use a thin-film detector structure to boost the signal-to-noise ratio of the output signal. Although building this detector adds a second critical masking level to the production process, the decrease in yield due to the additional step is expected to be more than offset by faster testing. As it turns out, testing tends to be a major part of the chip cost as the capacity reaches 4 Mbits, according to Dave Dossetter, bubble memory product marketing engineer.

Perhaps appropriately for a 4-Mbit memory, Intel worked with a manufacturer of lithography equipment and a mask maker to use X-ray lithography. Although contact printing was employed during development, Intel plans to put X-ray lithography to work for volume production, which would make it the first such commercial use.

A 4-Mbit bubble memory chip, supported by a full complement of six dedicated circuits, stands poised for applications ranging from industrial control to telecommunications to personal computers.

Bubble chip packs 4 Mbits into 1-Mbit space

Bubble memories sport a hefty list of advantages for mass storage applications. Yet because of the complexity of interfacing them, most designers have shied away from these devices, leaving them outcasts. But the sheer appeal of 4 Mbits tucked into a 20-pin package, coupled with a set of components that takes care of the complexities of linking a bubble chip to conventional host computers, makes an extremely attractive option for those designers who have previously resigned themselves to simpler but less attractive mass storage.

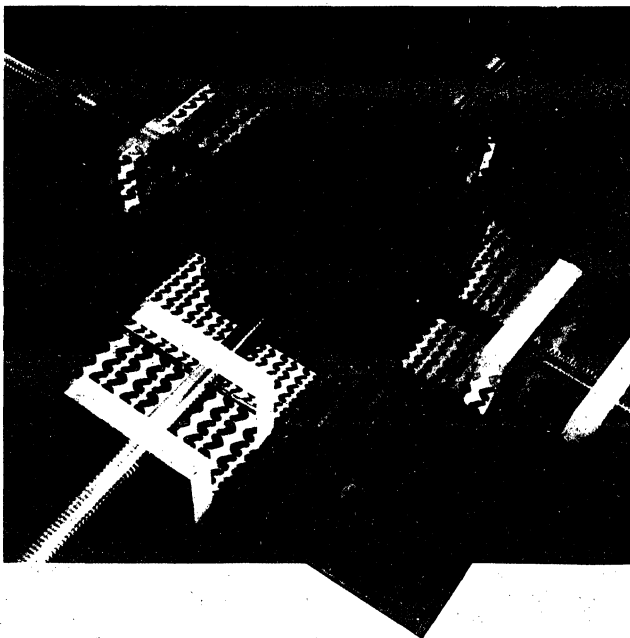
As for those who have already taken the plunge into bubbles with the chip's 1-Mbit predecessor, the 7110, upgrading to the 4-Mbit 7114 requires only minimal changes.

Some of those ready to benefit from a simplified bubble memory system are portable equipment makers, who will take advantage of the compactness and nonvolatility of bubble chips. Industrial control and robotics manufacturers will appreciate bubble devices' resistance to hostile environments, since they have no moving mechanical parts to succumb to shock, corrosion, or high humidity. These last three qualities also are important to telecommunications suppliers, who need low-cost, reliable buffers for PABX and other message-carrying systems.

Still, to reap the rewards inherent in bubble memories, a full complement of support circuits must accompany the bubble chip itself. Those companions are ready, in the form of

the 7224 bubble memory controller, the 7244 formatter and sense amplifier, the 7250 coil predriver, the 7254 VMOS driver transistor, and the 7234 current pulse driver.

Despite these components, a 4-Mbyte bubble memory system takes less space than the previous 1-Mbyte design, since the new bubble chip's package is both narrower, allowing more chips per board, and shorter, giving more room to stack boards next to one another (see "More Memory in Less Space"). Furthermore, the support components are interchangeable and, like the bubble chips, do not have to be matched sets, as was often true of other bubble devices. In fact, any bubble chip is guaranteed to



Hudson Washburn, Design Engineer
Sam Nicolino, Design Engineer
Intel Corp.
3065 Bowers Ave., Santa Clara, Calif. 95051

work with any support component, so that components can be replaced in the field without fine tuning.

Also, because the 4-Mbit bubble chip was designed to be compatible with the same hardware and software developed for the 1-Mbit version, the support circuits for both have the same pinouts. Most of the register bits are the same, too. The only differences are those in which the larger memory capacity affects how the bits are defined. Consequently, from a software perspective, any revisions to upgrade to the 4-Mbit chip are minor.

As with the 1-Mbit system, the user's interface with the 4-Mbit system remains simple. The software is written so that, first, parameters are passed to the controller by loading its registers, followed by commands. In addition, data is written or read in any of three transfer modes—DMA, polled, or interrupt—and the controller's 40-byte FIFO acts as a buffer between the host and formatter-sense amplifier chips. The formatter-sense amplifier is responsible for sending and receiving serial data

between the bubble and the controller. The host system therefore need only monitor the controller's status register to determine when it is busy and to see if a transfer operation was successful.

The bubble memory controller is the bubble chip's link to the host. It communicates with the host over an 8-bit bidirectional data bus; a single address line (A_0); and a chip-selection, a read and a write control, and an interrupt line. In addition, a ninth data bit line (D_8) can be used to detect parity errors.

The remaining input and output lines of the controller connect the formatter-sense amplifier, the coil predriver, and the current-pulse generator. These components, plus a pair of VMOS drive transistor chips, make up a 4-Mbit bubble storage unit (Fig. 1). Up to eight such units may be connected to a single controller, allowing users to trade off the number of pages against the individual page size to fit their data transfer requirements.

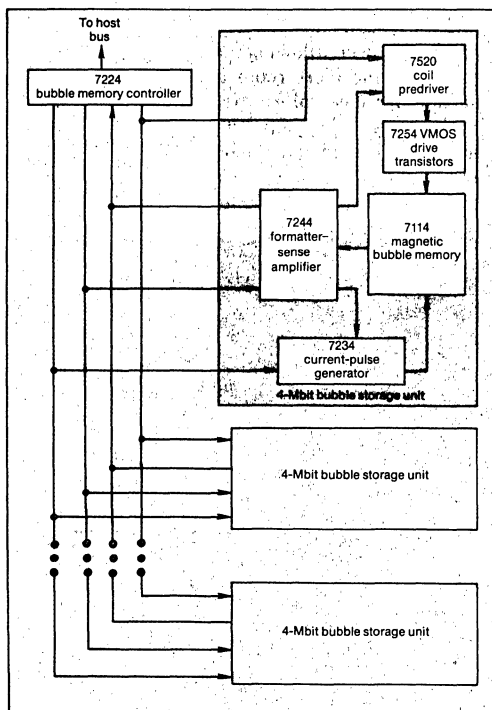
The controller close up

To understand the software and hardware interface with the bubble subsystem requires and understanding of the controller. An HMOS chip, it is housed in a 40-pin DIP and divided into 10 functional blocks (Fig. 2).

The host processor operates the bubble memory system by reading from, or writing to, specific registers within the bubble memory controller. The host selects each register by placing an address on lines A_0 and D_0 through D_4 . Specifically, the status register and command register are directly addressed using these six bits; a third register, the register address counter, is also directly addressed and in turn indirectly addresses the remaining registers, including the block-length register, the FIFO data buffer, and the enable register. These remaining registers are called parametric registers because they contain the flags and parameters that determine exactly how the controller will respond to commands written in the command register. The parametric registers are located in a register file and are selected with addresses 1011 through 1111. In general, the parametric registers must be loaded before commands are issued to the controller.

Parametric registers are loaded when they are addressed by the register address counter. The controller automatically increments the counter by one after each data transfer between the host and a parametric register. Thus there is no need to reload the address register in the case of multiple register reads and writes.

The address register increments, starting with the address first loaded, until it reaches binary address 1111. It then wraps around to 0000 and halts until it is reloaded with another address. However, when



1. The key to building a 4-Mbyte bubble memory system is the ability of the bubble chip's support ICs to simplify the interface with the host. Five such ICs plus a single 4-Mbit chip (shaded) form the basic memory block. Up to seven additional blocks in parallel, all governed by one memory controller chip, complete the system.

line A_0 is zero, all data transfers are with the FIFO. In addition, any other commands or a controlled stop sequence will reset the address counter to 0000, which is the FIFO address.

The most commonly used commands (see the table) are Initialize, Read Bubble Data, and Write Bubble Data. Others used in a typical operation are Read Seek, Write Seek, Read Formatter-Sense Amp Status, and Reset FIFO. In addition, two commands—Zero Access Read Seek and Zero Access Read Bubble Data—slash the data access time by a factor of more than 150. Zero Access Read Bubble Data returns the first byte of data in the FIFO within 50 μ s after the command is sent, provided the address is known in advance of the access command.

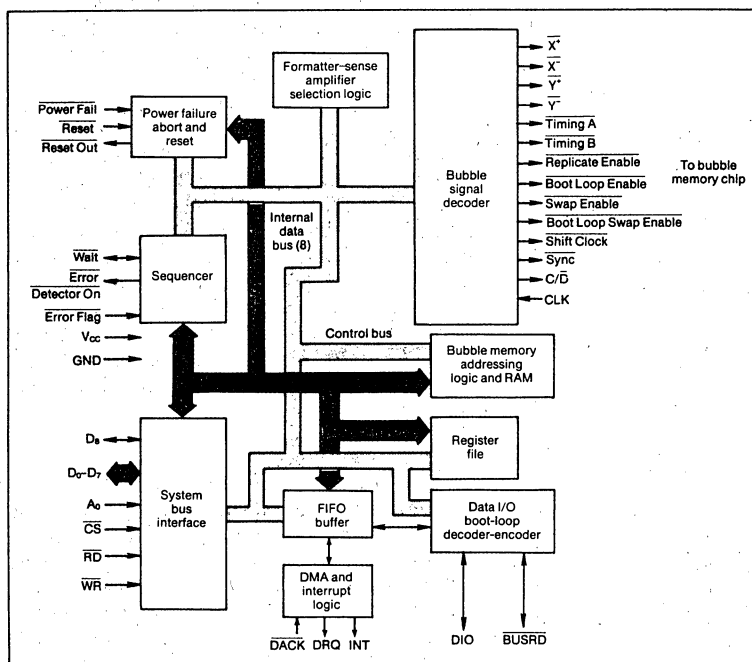
Parameters first

Commands are written by the host into an 8-bit write-once command register. Depending on the command, certain parameters must already be written into their respective registers. For example, the

Initialize command must be preceded by the number of formatter—sense amplifiers in the block-length register's first four MSB locations (Fig. 3a). Similarly, before issuing a Read Bubble Data command, the starting address information must already be set in the address register (Fig. 3b), as must be the number of system pages in the block-length register. Thus each command has its specific set of parametric requirements that must be established before it is issued.

If the parametric conditions have been set, the command is issued using a 5-bit command code. For example, Initialize is 00001, Read Bubble Data is 00010, and so on.

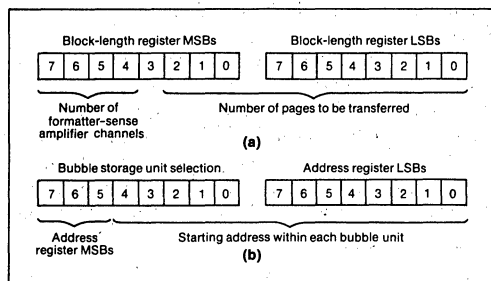
Information about any error condition, the completion or termination of a command, or the controller's readiness is stored in the status register. The host can directly address this register by setting the A_0 line and examining the eight status flags. The status register is updated every microsecond. Bits 1 through 6 (Fig. 4a) are set during command



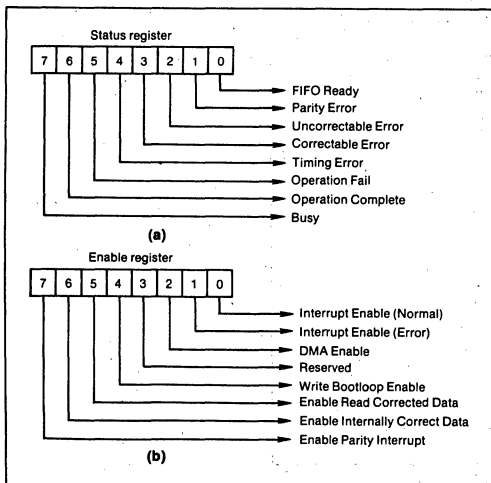
2. The 7224 bubble memory controller interfaces the bubble storage units with the host processor. It performs 10 functions, each represented by a block. The host is connected to an 8-bit data bus with an optional parity bit, a single address line, a chip-selection line, and a read and a write control line. Interrupt and DMA handshaking also are available.

execution and are reset when a new command is issued. The flags in the status register indicate whether the controller is executing a command or has completed one. In addition, they show whether an uncorrectable error or a timing error has occurred. Also, using a parity bit, the controller checks the data the host sends it and generates an odd parity for the data it sends to the host. Any parity errors are flagged.

The system page size and the number of pages to be transferred in response to a single bubble memory



3. The parametric registers set the basic conditions for transfers between the host and the bubble memory system. The block-length register gives the number of formatter-sense amplifier channels and the number of system pages in a block (a). The address register gives the starting address for a read or write command (b).



4. The status register bits (a) tell the host about any data errors, the state of the controller's readiness, or whether a command was completed properly or not. The register is updated every microsecond and indicates whether a data error was correctable or not, in addition to pointing out parity and timing errors. The enable register bits (b) specify several conditions, including interruption on an error, DMA enabling, and parity error interruption.

data read or write command are set by the block-length register, a 16-bit write-once register. The system page size is proportional to the number of bubble storage units operating in parallel during a data read or write operation. Each bubble chip requires two formatter-sense amplifier channels, with bits 4 through 7 specifying the number of such channels to be accessed. For example, in a 4-Mbyte system, if bits 7 to 4 are 0001, two channels will be accessed, each page will contain 512 bits, and there will be 65,172 pages. Setting the bits to 0100 specifies eight channels, 2048 bits per page, and 16,384 pages.

The right address

Which bubble memory group is accessed and what the starting address location is within that group are determined by the contents of the address register. Each bubble chip has 8192 address locations for reading or writing data. Consequently, 13 bits are needed to specify an individual bubble storage unit's starting address. Which of the units to be read from or written to is indicated by address register bits 5 through 7. How the controller interprets these bits depends on the number of bubble storage units in a group as specified by the block-length register. For example, if the formatter-sense amplifier channels are numbered 0 through F_{16} and the number of formatter channel bits of the block-length register are set at 0000, the address register bits will specify channels 0 through 7. If, on the other hand, the block-length register bits are in the sequence 0001, the address register bits select the formatter-sense amplifier channel pairs and address register bits 0110 select channels C and D.

The address range for a 4-Mbyte subsystem is 0000-FFFF, or 65,172 pages. Selecting address register bits 0111 puts the data in the last 8192 pages of bubble storage.

Enable register controls

Certain functions in the formatter-sense amplifier and the controller are governed by setting bits in the enable register (Fig. 4b). For example, setting the Enable Parity Interrupt stops the host when the controller detects a parity error on the data bus lines (D_0-D_7). Also, the controller operates in a DMA data transfer mode when the DMA Enable bit is set. In this mode the Data Request and Data Acknowledge interface signals become operational; otherwise, the controller supports interrupt-driven or polled data transfer modes. As a result, users have a choice of three data transfer methods.

The Interrupt Enable (Normal) bit, when set to a 1, allows the controller to interrupt the host system when a command is successfully executed. The Interrupt Enable (Error) bit works in conjunction with

Bubbles by the block

The basic technology of the 7114 4-Mbit bubble chip—known as field access, conductor-first permalloy—is the same as used to build the earlier 7110, a 1-Mbit part, except for several important refinements. These refinements quadruple the bit density and the data transfer rate.

The increased density is produced by halving the period of the basic memory cell (called an asymmetric propagator) to 5.5 μm . The resultant chip size is 501 by 580 mils (compared with the 1-Mbit's 512 by 614 mils). A 0.75- μm minimum feature size, smaller than that of any silicon chip, is being printed now in development volumes using optical contact lithography. However, X-ray lithography techniques will be used for production volumes to achieve repeatable results despite the small minimum-feature size.

In addition, a thin-film detector was developed that doubles the detected bubble signal compared with the previous thick-film detectors. This makes doubling the data rate feasible. Further, doubling the field rotation rate from 50 to 100 kHz also doubled the data rate,

producing the overall 400% increase, which also means an average random access time of 40 ms. (A 50-kHz version will be introduced first that has twice the data rate of the 1-Mbit chip and an 80-ms access time.)

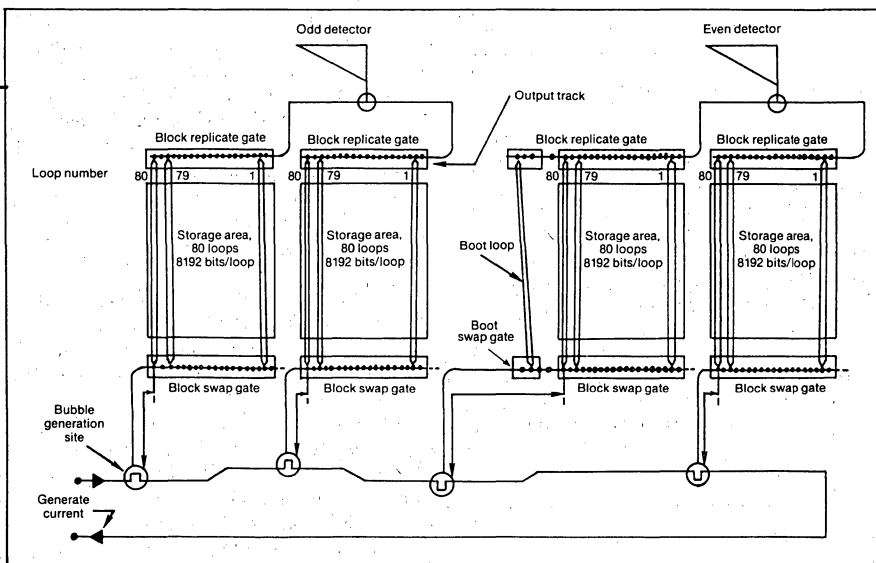
Like the technology, the architecture of the 4-Mbit chip is an enhanced version of the 1-Mbit design. Both use block-swapping and replicating schemes to write and read bubbles in parallel, to ensure nonvolatile storage, and to permit the use of multiplexed replication generators to reduce the number of external pins.

The page length is fixed at 512 bits (64 bytes), but the number of pages has been quadrupled for the 4-Mbit part. Both chips are organized into identical halves. Thus, from an architectural perspective, the higher-density chip looks like a 1-Mbit part with four times the number of pages and either twice (50 kHz) or four times (100 kHz) the data rate.

Actually, the 7114 is divided into eight octants, each comprising 80 minor loops, and each loop containing 8192 bits (see the figure). The 7110, in comparison, is split

into four quadrants, each with 80 minor loops, but each loop contains only 4096 bits. Also, whereas the 7110 was designed to sense one bit per side per field rotation, the 7114 senses two bits. In the 50-kHz 4-Mbit part, the longer loops are compensated for by the two-bit-per-rotation sensing.

Like the 1-Mbit device, the 4-Mbit chip has redundant loops to ensure a high yield of devices with the full 4,194,304 bits of storage capacity. Redundancy increases yields and so lowers device cost. During manufacture, each device is individually tested and a record of faulty loop locations is written and stored in the device's bootstrap loop, known as the "boot loop." The boot loop's contents are used by the 7224 bubble memory controller during initialization, reading, and writing to provide a full 4-Mbit memory space to the user while keeping redundant loops invisible. The major-track, minor-loop architecture used by both the 7114 and the 7110 to accomplish the writing, reading, and nonvolatile storage of data also maintains the reliability inherent in bubble technology.



the other enable register bits to support three levels of error correction.

At the first level, setting Enable Internally Correct Data causes the controller to send a command to a formatter—sense amplifier when an error has been detected. The formatter—sense amplifier responds by internally cycling the data through its error-correction network. On completion, it sends its status to the controller, indicating whether or not the error was corrected.

For the second level, the Enable Read Corrected Data bit prompts the controller to issue a command to the appropriate formatter—sense amplifier when an error has been detected. The formatter—sense amplifier then corrects the error if possible and transfers the corrected data to the controller. When

the data transfer is complete, the controller reads the formatter—sense amplifier's status to determine whether the error was corrected. Otherwise, faulty data could be transferred to the controller and possibly to the host.

Lastly, setting the Write Bootloop Enable bit permits writing into the bootstrap loop, called here just the "boot loop." Normally, the loop should only be read, but under special circumstances a user may wish to write into it.

The FIFO as a data buffer

All data moving between the host and the bubble units passes through the 40-byte FIFO buffer. As a result, the data transfer is asynchronous, with timing constraints relaxed somewhat for both the formatter—sense amplifier and the host system. When the controller is busy executing a command, the FIFO functions as a data buffer; however, when the controller is not busy, the FIFO is available to the host as a general-purpose FIFO register bank.

Actually, a total of 43 bytes of data may be stored in the controller: 40 bytes in the FIFO, 1 byte each in its input and output latch, and 1 byte in the controller's input latch. During execution of a command involving a data transfer between the host and the formatter—sense amplifiers, the data passes through the FIFO and its status is indicated by the FIFO Ready bit in the storage register.

The FIFO is addressed automatically after the last parametric register has been written into; alternatively, the host can explicitly address the FIFO by writing the address 0000 into the register address counter. Also, after a Write Bubble Data, a Write Boot-Loop Register, or a Write Boot-Loop Register Masked command is issued, the controller delays the data transfer until there are at least two bytes of data in the FIFO. Furthermore, it is the host system's responsibility to keep up with the data transfer during execution of a command; otherwise the FIFO could underflow or overflow. If either case occurs, a Timing Error bit is set in the status register.

A look at data transfer

The boot-loop register plays a key role in data transfer both for writing and reading. This 160-bit register contains information detailing the configuration of good and bad loops in the corresponding channel of each bubble chip.

Each bit of the register corresponds to a minor loop in the bubble chip. As data passes through the latter's I/O latches, the contents of the boot-loop register are used during reading to remove the bits corresponding to bad loops and during writing the contents are used to insert 0s in those bit positions that correspond to bad loops.

Bubble controller command codes

D ₇	D ₆	D ₅	D ₄	D ₃	Command name
0	0	0	0	0	Write—Boot Loop Register Masked
0	0	0	0	1	Initialize
0	0	0	1	0	Read Bubble Data
0	0	0	1	1	Write Bubble Data
0	0	1	0	0	Read Seek
0	0	1	0	1	Read Boot Loop Register
0	0	1	1	0	Write Boot Loop Register
0	0	1	1	1	Write Boot Loop
0	1	0	0	0	Read Formatter—Sense Amp Status
0	1	0	0	1	Abort
0	1	0	1	0	Write Seek
0	1	0	1	1	Read Boot Loop
0	1	1	0	0	Read Corrected data
0	1	1	0	1	Reset FIFO
0	1	1	1	0	Memory Unit Purge
0	1	1	1	1	Software Reset
1	0	0	1	0	Zero Access Read Bubble Data
1	0	1	0	0	Zero Access Read Seek

More memory in less space

Instead of a leadless package requiring a second, leaded socket, the 7114 4-Mbit bubble chip is housed in a leaded package that can be placed in a socket or soldered directly to a PC board. Like the 1-Mbit package, it has 20 pins. However, the distance between pin rows is smaller, making the footprint smaller and allowing designers to incorporate more components onto the board. Also because the package's height is smaller, boards can be spaced as close as 0.6 in. to one another. Thus consequently, either more boards can be accommodated or the overall system size can be made smaller. As a result, a 4-Mbyte bubble memory system can be built in less space than a 1-Mbyte bubble system.

Meanwhile, the error-correction block implements a 14-bit Fire code error-detection and -correction process. If it has been enabled by the user, the error-correction circuitry appends the 14-bit code to the end of each 256-bit block of data that passes through the FIFO during a data write operation. When data is being read, this circuitry checks the data block and notifies the controller with an error flag when an error has been detected.

As stated earlier, a Write Bubble Data command from the controller to the formatter-sense amplifier permits data from the controller to be written into the good loops of the memory unit. If the error correction is activated, the amplifier automatically adds the 14 error-correction bits to the end of each 256-bit data block.

Similarly, a Read Bubble Data command enables the formatter-sense amplifier to read data from the bubble chip, as was also mentioned previously. This data is sensed by the sense amplifiers and screened by the boot-loop registers so that only data from good loops is written into the FIFOs. If the error correction is selected, data to be read is first buffered. That is, a full block (270 bits) of data is collected in the FIFO before any bits are read out. As a result, the

error-correction circuitry detects any errors and interrupts the controller before any data is sent. If there are no errors, the 270-bit block is read from the FIFO and sent to the controller while the next block is loaded into the FIFO.

In contrast, an Internally Correct Data sequence forces the formatter-sense amplifier to cycle the data internally through the error-correction network without sending any of it to the controller. At the end of the operation, the amplifier sets a Correctable or Uncorrectable Error bit in its status register. If the error is correctable, the controller has the option of issuing a Read Corrected Data command. This command cycles the data through the error-correction circuitry as it is being read by the controller. After all 256 bits have been transferred to the controller, the formatter-sense amplifier status register indicates whether the error was found to be correctable or not. The Read Corrected Data command is used even when the data has been previously corrected by the Internally Correct Data command. □

The authors wish to thank Dave Dossetter, Product Marketing Engineer, and Dick Pierce, Marketing Applications Engineer, for their invaluable assistance in preparing this article.